



Un algorithme efficace pour la mise en correspondance des segments 2D dans une séquence d'images

Ming Xie, Patrick Rives

► To cite this version:

Ming Xie, Patrick Rives. Un algorithme efficace pour la mise en correspondance des segments 2D dans une séquence d'images. [Rapport de recherche] RR-0929, INRIA. 1988. inria-00075627

HAL Id: inria-00075627

<https://inria.hal.science/inria-00075627>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNITÉ DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

N° 929

Programme 6

UN ALGORITHME EFFICACE POUR LA MISE EN CORRESPONDANCE DES SEGMENTS 2D DANS UNE SEQUENCE D'IMAGES

**Ming XIE
Patrick RIVES**

Novembre 1988



★ R R - 8 9 2 9 ★

Campus Universitaire de Beaulieu
35042 - RENNES CÉDEX
FRANCE
Téléphone: 99 36 20 00
Télex: UNIRISA 950 473 F
Télécopie: 99 38 38 32

Un algorithme Efficace Pour La Mise en Correspondance Des Segments 2D Dans Une Séquence d'Images

===

An Accurate Algorithm For The Matching Of 2D Line Segments Within A Sequence Of Images

Ming XIE and Patrick RIVES
INRIA/IRISA de Rennes

Publication Interne n°430 - Septembre 1988 - 36 Pages

Campus de Beaulieu
F-35042 Rennes Cedex FRANCE
Tel (33)-99-36-20-00

Résumé

La mise en correspondance des segments 2D dans une séquence d'images acquises par un capteur monoculaire ou polynoculaire pose un problème délicat soit pour la reconstruction des segments 3D soit pour l'estimation du mouvement, basée sur l'emploi des segments 2D. Nous présentons dans cet article une approche robuste pour résoudre ce problème. Dans une séquence d'images, nous travaillons à chaque instant sur deux images voisines : l'image courante et l'image suivante. L'idée de base de notre méthode consiste à : (a) extraire les contours spatiaux sur l'image courante en vue de localiser les positions des contours, (b) effectuer le chaînage des contours et l'approximation polygonale des chaînes en vue d'estimer précisément les orientations

des contours, (c) estimer les composantes de vitesse perpendiculaires aux contours, (d) calculer les paramètres dynamiques (valeurs statiques avec leurs dérivées) des segments 2D et (e) suivre les indices des segments 2D le long de la séquence d'images.

Dans notre approche, l'estimation des déplacements des contours entre deux images voisines est réalisée par l'emploi des masques présentés dans [4]. A l'étape de l'approximation polygonale d'une chaîne, nous modélisons un segment 2D par ses coordonnées *plückeriennes*. Ceci nous permet d'estimer directement les paramètres d'un segment 2D sur l'image courante et les paramètres de son correspondant sur l'image suivante.

Mots Clés : Contour Dynamique, Mise en correspondance des primitives, Fonction de densité jointe de probabilité, Chaînage des contours, Coordonnées *Plückeriennes*.

Abstract

The matching of 2D line segments within a sequence of images is a key point for the problems of : (a) 3D line segment reconstruction from monocular or polynocular stereovision and (b) motion estimation from 2D line segment correspondence. The 2D line segments can be accurately matched by the method presented in this paper. Within a sequence of images, at each instant, we work on two consecutive images : the present image and the next image. The basic idea of our method consists of : (a) the extraction of static contours in the present image in order to localize the positions of edgels, (b) the edge linking and polygonal approximation in order to accurately estimate the directions of the edgels, (c) the estimation of the edgels' normal optic flow between two consecutive images, (d) the calculation of dynamic parameters (static values with their derives) of 2D line segments and (e) the following of index of 2D line segments within a sequence of images.

In our method, the estimation of edgels' optic flow is realized with the help of templates presented in [4]. At the step of approximating a chain by the 2D line segments, we use *plücke* coordinates to represent a line segment. Because of this representation, we can calculate directly the parameters of a 2D line segment in the present image and that of its correspondence in the next image.

Key Words : Dynamic contour, Matching of primitives, Joint density function of probability, Linking of contour, *Plücke* coordinates.

Table des matières

1	Introduction	5
1.1	Contexte	5
1.2	Description Du Problème	7
2	Extraction Des Contours Statiques	11
3	Approximation Polygoneale Des Contours	12
4	Estimation Des Déplacement Des Contours	16
5	Mise En Correspondance Des Segments 2D	21
6	Suivi Des Indices Des Segments 2D	24
7	Résultats Expérimentaux	27
8	Conclusion et Discussion	29

1 Introduction

La vision par ordinateur est une discipline qui tente de développer sur des machines informatiques des algorithmes ayant pour but de percevoir et d'interpréter un environnement 3D afin d'en extraire les caractéristiques nécessaires à la résolution d'un problème donné. L'étape de perception consiste à extraire l'information sensorielle. L'étape d'interprétation consiste à établir la liaison entre l'information sensorielle et une description symbolique du modèle de cet environnement. Le résultat de l'interprétation permet au système décisionnel d'exécuter les actions nécessaires au bon accomplissement d'une tâche donnée.

Il y a deux types de stratégies utilisées afin d'atteindre les buts de la vision par ordinateur : ascendantes et descendantes. Les stratégies ascendantes tentent de construire à partir de l'information sensorielle (primitives 2D) une représentation de plus haut niveau (ex: primitives 3D) interprétable par le processus décisionnel. Les stratégies descendantes tentent de déduire à partir d'un ensemble d'objets connus par le système une description compatible avec les primitives extraites de l'image, le processus décisionnel étant alors basé sur une mesure de distance entre la représentation extraite de l'image et la description issue du modèle. En fait, la combinaison de ces deux stratégies dans un système de vision constitue une approche plus sophistiquée.

1.1 Contexte

Une des tendances récentes de recherches en vision par ordinateur consiste à prendre en compte une séquence d'images au lieu d'avoir une ou deux images dans le cas de vision statique. Cette approche est connue sous le nom de "vision dynamique" (si le mouvement est contrôlable, on l'appelle aussi "vision active") et possède deux avantages fondamentaux :

1. Certains problèmes rencontrés (par exemple: Shap from shading) en vision par ordinateur peuvent être résolus à partir d'équations linéaires. La solution trouvée est unique et stable.

2. La possibilité d'utiliser la cohérence temporelle au sein de la séquence d'images permet de raffiner les estimation des primitives par des méthodes de filtrage (par exemple : Filtre de KALMAN).

Si nous considérons le cas particulier des robots mobiles, l'utilisation de la vision dynamique procure d'autres avantages :

1. Premièrement, elle donne la possibilité au robot d'acquérir en ligne l'information sur son environnement de façon à pouvoir exécuter correctement sa tâche dans un univers inconnu ou mal modélisé.
2. Deuxièmement, elle donne la possibilité de réaliser une commande en boucle fermée qui permet d'avoir :
 - (a) La capacité de contrôler le mouvement de caméra en vue d'améliorer la qualité d'estimation de primitives 3D (vision active).
 - (b) La capacité de compenser l'incertitude sur les primitives (estimées) par une commande en boucle fermée ayant de bonne propriétés de robustesse et de stabilité.

Nous avons focalisé nos recherches sur l'utilisation d'une caméra embarquée sur un robot et d'algorithmes de vision dynamique (ou active). Notre objectif de recherche se définit de la façon suivante :

En supposant que le robot se déplace avec une vitesse contrôlable dans un environnement inconnu composé d'objets polyédriques, essayer de reconnaître cet environnement à partir de la séquence d'images observées par une caméra embarquée sur le robot.

Nous travaillons dans le cadre de la vision monoculaire. Puisque l'application de notre recherche est restreint à un univers polyédrique, nous portons naturellement attention aux primitives de type : segments 2D et segments 3D qui sont les types de primitives plus distinctifs. Nous avons proposé un système de vision qui répond à ce besoin. Les aspects principaux de ce systèmes sont :

1. Estimation des segments 2D en mouvement (ou dynamiques).
Il s'agit d'intégrer des techniques d'extraction de contours statiques, d'extraction de contours spatio-temporels et de chaînage des contours

afin d'estimer les segments 2D dynamiques le long d'une séquence d'images.

2. Commande en boucle fermée.

En vue d'améliorer la qualité de l'estimation des segments 2D dynamiques, on génère un mouvement local selon un critère de minimisation des erreurs d'estimation. Ce mouvement local est superposé au mouvement global de façon à obtenir un mouvement exécutable.

3. Estimation récursive des segments 3D.

Connaissant le mouvement de la caméra, il est possible de calculer directement les segments 3D à partir des segments 2D dynamiques selon une méthode générale que nous avons étudié. Cette méthode s'applique aussi à la vision polynoculaire.

4. Identification des polygones 3D.

On essaye d'extraire les polygones 3D interprétables à partir d'un ensemble des segments 3D estimés. Ceci permet de reconnaître et de localiser les objets polyédriques.

La figure 1 montre schématiquement notre système de vision dans le cadre de la reconstruction d'une scène 3D polyédrique. Nous voulons que chaque niveau de traitement dans notre système de vision donne un résultat robuste. Ceci est le point clef pour assurer la meilleure performance d'un système de vision.

1.2 Description Du Problème

La mise en correspondance des segments 2D dans une séquence d'images acquises par un capteur monoculaire ou polynoculaire pose un problème délicat soit pour la reconstruction des segments 3D soit pour l'estimation du mouvement, basée sur l'emploi des segments 2D. Nous présentons dans cet article une approche robuste pour résoudre ce problème. Dans une séquence d'images, nous travaillons à chaque instant sur deux images voisines : l'image courante et l'image suivante. L'idée de base de notre méthode consiste à : (a) extraire les contours spatiaux sur l'image courante en vue de localiser les positions des contours, (b) effectuer le chaînage des contours

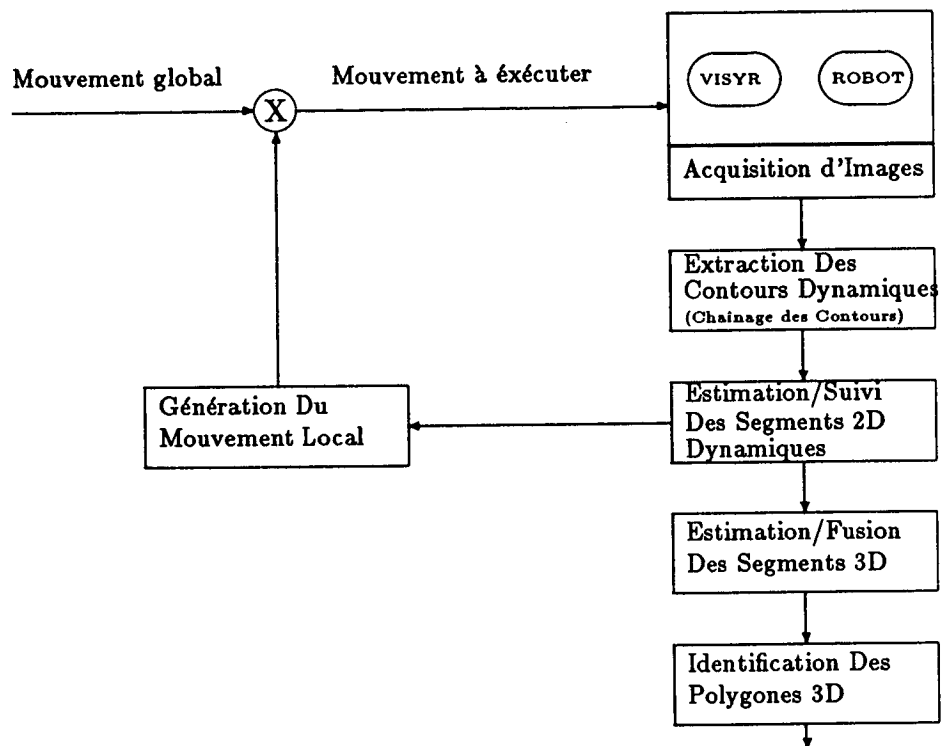


Figure 1 : Un Système pour la Reconstruction d'une Scène 3D Polyédrique

et l'approximation polygonale des chaînes en vue d'estimer précisément les orientations des contours, (c) estimer les composantes de vitesse perpendiculaires aux contours, (d) calculer les paramètres dynamiques (valeurs statiques avec leurs dérivées) des segments 2D et (e) suivre les indices des segments 2D le long de la séquence d'images.

Dans notre approche, l'estimation des déplacements des contours entre deux images voisines est réalisée par l'emploi des masques présentés dans [4]. A l'étape de l'approximation polygonale d'une chaîne, nous modélisons un segment 2D par ses coordonnées *plückeriennes*. Ceci nous permet d'estimer directement les paramètres d'un segment 2D sur l'image courante et les paramètres de son correspondant sur l'image suivante, c'est-à-dire :

$$\begin{aligned} &\text{les coordonnées plückeriennes: } (m, l) \\ &+ \\ &\text{leurs dérivées: } (\dot{m}, \dot{l}). \end{aligned}$$

En résumé, nous allons traiter d'abord le problème suivant :

Etant données deux images consécutives issues d'une séquence d'images, estimer les paramètres des segments 2D sur l'image courante et prédire ceux des leurs correspondants sur l'image suivante.

Le schéma présenté par la figure 2 résume notre stratégie de l'estimation des segments 2D sur deux images consécutives.

Ayant résolu ce problème, le suivi des segments 2D le long d'une séquence d'images est devenu une tâche triviale. Sur chaque image, on connaît à la fois les segments 2D estimés à l'instant courant mais aussi les segments 2D prédits à l'instant précédent. Dans le cas idéal où il n'y a pas d'erreurs d'estimation, ces deux ensembles des segments 2D sont superposés parfaitement. De ce fait, la mise en correspondance entre les segments estimés et les segments prédits sur une même image consiste à trouver le plus proche voisin pour chaque segment parmi ces deux ensembles.

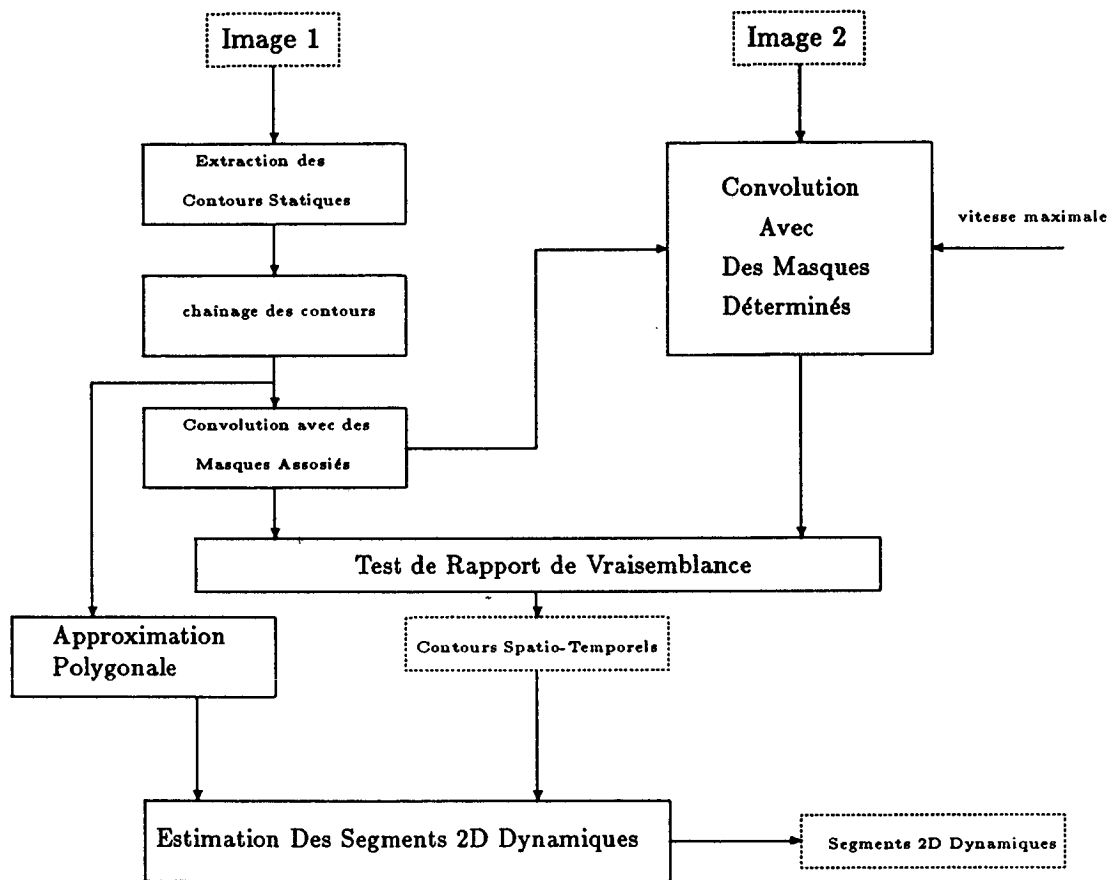


Figure 2 : Schéma d'Estimation des Segments 2D dynamiques

2 Extraction Des Contours Statiques

La technique de mise en œuvre l'algorithme d'extraction des contours spatio-temporels dans [4] consiste à prendre en compte à chaque instant deux images consécutives et à effectuer la convolution de ces deux images avec les masques prédéfinis. A la fin on détermine les composantes de vitesse associées aux contours par le biais d'un test de vraisemblance.

Il faut souligner que l'opération de la convolution est très coûteuse en temps de calcul, surtout quand il y a un grand nombre de masques. Cela conduit à réduire l'opération de la convolution le plus possible. En réalité, nous n'avons pas besoin d'effectuer la convolution pour tous les pixels d'image si nous pouvons localiser les pixels contour dans la première image par une méthode d'extraction de contours spatiaux (ou statiques).

A l'heure actuelle l'algorithme présenté dans [5] pour extraire les contours statiques représente une méthode plus robuste et plus rapide. L'estimation des gradients dans deux directions (horizontale(x) et verticale(y)) sur une image est réalisée par une convolution avec une fonction à réponse impulsionnelle infinie $f(s)$ et sa fonction d'intégration $g(s)$, c'est-à-dire :

$$\begin{cases} \text{gradient}(x) = g(y) \otimes f(x) \otimes \text{IMAGE}(x, y). \\ \text{gradient}(y) = g(x) \otimes f(y) \otimes \text{IMAGE}(x, y). \end{cases}$$

Le choix d'une fonction à réponse impulsionnelle infinie permet de transformer l'opération de convolution en un processus de filtrage récursif du style :

$$\begin{cases} F(Z) = \frac{a \bullet Z^{-1}}{1 + b_1 \bullet Z^{-1} + b_2 \bullet Z^{-2}}. \\ G(Z) = \frac{c_0 + c_1 \bullet Z^{-1}}{1 + d_1 \bullet Z^{-1} + d_2 \bullet Z^{-2}}. \end{cases}$$

Dans le cas de notre expérimentation, nous avons choisi :

$$\begin{cases} f(s) = -c \bullet e^{-\alpha \bullet |s|} \bullet \sin(w \bullet |s|). \\ g(s) = k \bullet \frac{\alpha \bullet \sin(w \bullet |s|) + w \bullet \cos(w \bullet |s|)}{\alpha^2 + w^2} \bullet e^{-\alpha \bullet |s|}. \end{cases}$$

L'utilisation de cet algorithme en vue de localiser préalablement les pixels contour dans la première image permet de réduire largement le nombre d'opérations de convolution avec les masques. Pour réduire davantage ce nombre, nous nous contentons d'effectuer en avance le traitement de chaînage des contours ce qui permet d'éliminer les pixels contours isolés (bruits) et d'estimer précisément l'orientation des contours.

3 Approximation Polygonale Des Contours

L'extraction de l'information concernant la structure des formes présentées dans une image est un problème très délicat. Bien qu'il ait été étudié depuis longtemps, il n'y a pas de méthode qui soit en même temps robuste, rapide et générale.

Nous avons étudié un algorithme efficace qui extrait tous les segments droits et leurs connexités en deux passes sur une image contour :

- Première passe : transformation de la structure des données de l'image contour.
- Deuxième passe : chaînage des contours et approximation polygonale.

Dans la première passe, on a introduit d'abord l'idée de "couple contour" qui est défini comme un sous-ensemble de pixels contour alignés consécutivement sur une même ligne. Ensuite on a redéfini une structure de base des données pour organiser les pixels contours qui comporte :

1. le tableau MCOUPLE de dimension $nbcouple \times 2$ ("nbcouple" est le nombre des couples contour dans l'image).
2. le tableau FDR (Fonction De Répartition des couples contour sur les lignes d'image) de dimension $n \times 1$ ("n" est le nombre des lignes de l'image).

L'opération de transformation de structure des données nécessite un temps d'exécution négligable. Par contre, il permet le chaînage des contours d'effectuer en une seule passe sur le tableau FDR.

Pour le chaînage des contours, on a d'abord défini une zone de voisinage causale pour chaque couple contour. Le choix de cette définition provient du fait qu'une zone de voisinage causale possède une propriété intéressante assurant la robustesse du chaînage. De plus, la taille de zone de voisinage n'est pas fixée, ce qui rend l'algorithme de chaînage très souple, adapté à toutes sortes d'images contours. On distingue quatre types des chaînes dans la zone de voisinage d'un couple contour. Selon la nature des chaînes présentes, l'opération de chaînage de notre algorithme consiste à faire : (a) la création des chaînes, (b) le rallongement des chaînes et (c) la fermeture des chaînes.

Le traitement de chaînage assure une simplicité totale des chaînes créées, cela signifie que chaque chaîne n'a aucune branche secondaire (bifurcation). En plus les couples contour d'une même chaîne sont positionnés dans le tableau MCOUPLE de façon séquentielle en ordre croissante par rapport à X et à Y. Cela permet de réaliser directement la fraction des chaînes en morceaux de segments droits.

Etant donné une chaîne, on prédit d'abord un segment droit en utilisant le couple tête et le couple queue de cette chaîne. Ensuite, on teste la nécessité de l'opération de fraction. S'il y a au moins un couple contour qui s'éloigne de ce segment prédit à un seuil donné, on découpe cette chaîne au couple contour qui s'éloigne le plus et on obtient deux sous-chaînes. Cette opération va continuer jusqu'à ce qu'il n'y ait plus de fraction à faire. L'ensemble des segments droits prédits qui n'ont pas causés la fraction des chaînes constitue donc une approximation polygonale de la chaîne initiale. La figure 3 illustre l'approximation d'une chaîne.

Après l'opération de fraction des chaînes, l'ensemble des couples contour appartenant à une chaîne est partitionnés en sous-ensembles. Chaque élément de ce sous-ensemble forme un segment droit dont on peut estimer les paramètres.

Un segment droit 2D est modélisé en principe dans un espace 2D par le vecteur normal perpendiculaire au segment et la distance à l'origine de ce segment.

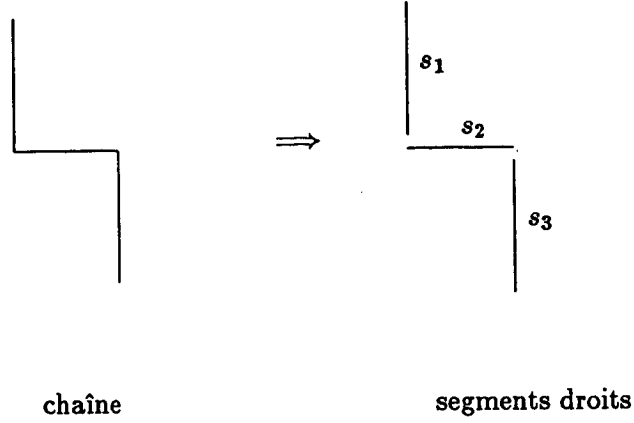


Figure 3 : Approximation Polygonale d'une Chaîne

Soit $\vec{n} = (a, b)$ le vecteur normal du segment S et c sa distance à l'origine, $f = (a, b, c)$ constitue donc le vecteur paramètre du segment S et son équation s'écrit comme :

$$a \bullet x + b \bullet y - c = 0.$$

où (x, y) est un point sur le segment S .

Etant donné un sous-ensemble de m couples contour $\{(x_i, y_i); i = 1, m\}$ où (x_i, y_i) est le point central du couple contour i , la meilleure estimée du vecteur f sera réalisé par optimisation d'un critère tel que :

$$J_f = \min_f \{ J(f) = \sum_{i=1}^m (a \bullet x_i + b \bullet y_i - c)^2 \}$$

avec $a^2 + b^2 = 1$.

La minimisation du critère J_f revient à trouver le vecteur paramètre f satisfaisant les équations suivantes :

$$\frac{\partial J(f)}{\partial f} = 0.$$

C'est-à-dire :

$$\begin{cases} p \bullet a + q \bullet b - s \bullet c = 0 \\ q \bullet a + w \bullet b - t \bullet c = 0 \\ s \bullet a + t \bullet b - m \bullet c = 0 \\ a^2 + b^2 = 1 \end{cases} \quad (1)$$

où :

$$p = \sum_{i=1}^m x_i^2. \quad q = \sum_{i=1}^m (x_i y_i). \quad w = \sum_{i=1}^m y_i^2.$$

$$s = \sum_{i=1}^m x_i. \quad t = \sum_{i=1}^m y_i.$$

Après quelques développements mathématiques, nous trouvons la solution qui est :

- Cas 1 : segment droit horizontal.

$$f = \begin{pmatrix} 0 \\ 1 \\ t \end{pmatrix}.$$

- Cas 2 : segment droit vertical.

$$f = \begin{pmatrix} 1 \\ 0 \\ s \end{pmatrix}.$$

- Cas 3 : segment droit en orientation quelconque.

$$f = \begin{pmatrix} \frac{w \bullet s - q \bullet t}{L} \\ \frac{p \bullet t - q \bullet s}{L} \\ \frac{p \bullet w - q^2}{L} \end{pmatrix}; \text{ avec } L = \sqrt{(w \bullet s - q \bullet t)^2 + (p \bullet t - q \bullet s)^2}.$$

En fonction des orientations des pixels contours, on choisit les masques de convolution adaptés pour l'estimation des déplacements des ceux-ci.

4 Estimation Des Déplacement Des Contours

Une étude fondamentale dans le domaine de la vision dynamique est l'extraction d'informations spatio-temporelles, à savoir l'extraction du champs des vitesses le long d'une séquence d'images numériques ordonnées dans le temps. Un cas particulier d'estimation du champs des vitesses consiste à estimer le champ des vitesses restreint aux contours images. Les attributs estimés sont :

1. leur position (x_0, y_0) dans l'image.
2. leur direction spatiale locale θ dans l'image.
3. la composante du vecteur vitesse associé v^\perp perpendiculaire au contour.

Nous pouvons considérer une courbe de contour dans une séquence d'images comme une surface 3D dans un espace 3D formé de deux dimensions spatiales et d'une dimension temporelle. Géométriquement, la surface 3D peut être approximée localement par une portion de surface plane, et photométriquement les deux régions adjacentes séparées par la surface 3D peuvent être considérées localement comme ayant une intensité lumineuse constante.

Nous donnons par la suite un rappel du principe de la méthode présentée dans [4] en vue d'illustrer la propriété des masques de convolution que nous avons utilisée pour estimer les composantes de vitesse perpendiculaires aux contours. La méthode citée s'appuie sur la technique de modélisation et met en œuvre une technique de test de vraisemblance.

Modélisation d'Un Élément De Contour En Mouvement (ECM)

Une séquence d'images est considérée comme étant un espace à trois dimensions (x, y, t) : deux dimensions spatiales et une dimension temporelle. Dans cet espace, un élément de contour en mouvement (ECM) génère une portion de surface. A défaut de toute information complémentaire, on considère cette portion de surface comme étant planaire. Ce choix s'apparente

à une approximation du premier ordre.

Dans l'espace (x, y, t) , la portion de surface d'un ECM est donc engendrée au sein d'un volume π par :

- l'élément de contour spatial référencé au point (x_0, y_0) et représenté par un petit segment droit dans le plan $t = t_0$.
- son vecteur de vitesse (ou vecteur de déplacement) associé $V = (dx/dt, dy/dt, 1) = (v, 1)$.

Hypothèses De Configuration d'Un ECM

Au préalable, il est nécessaire de préciser les deux hypothèses de configuration locale concernant un ECM. Etant donné un volume π placé au point (x_0, y_0) dans l'espace (x, y, t) , deux hypothèses peuvent intervenir :

1. Hypothèse H_0 :
il n'existe pas d'ECM dans π ; alors, la distribution des intensités au sein de π est modélisée selon :

$$\text{niveau constant } c_0 + \text{bruit}$$

où le bruit est un bruit Gaussien centré de variance σ^2 .

2. Hypothèse H_1 :
il existe un ECM dans π ; alors, cela se traduit par la présence d'une portion de plan S divisant le volume π en deux sous-volumes π_1 et π_2 où les niveaux constants des modèles d'intensité, du même type que précédemment, sont respectivement c_1 et c_2 , avec $c_1 \neq c_2$.

Test De Vraisemblance Pour La Détermination Des ECMs

On définit une fonction de vraisemblance pour chacune des hypothèses. Cette fonction est en fait la densité jointe des lois de probabilité des intensités aux points du volume π , qui suivent des lois gaussiennes, c'est-à-dire :

- $N(c_0, \sigma^2)$ en tout point de π pour l'hypothèse H_0 .
- $N(c_1, \sigma^2)$ (resp. $N(c_2, \sigma^2)$) en tout point de π_1 (resp. π_2) pour l'hypothèse H_1 .

Pour l'hypothèse H_0 , la fonction de vraisemblance L_0 est calculée de la façon suivante :

$$L_0 = (2\pi\sigma^2)^{-\frac{n}{2}} \prod_{p_i \in \pi; i=1}^n \exp\left\{ -\frac{1}{2\sigma^2} (f_{p_i} - c_0)^2 \right\}. \quad (2)$$

où $\{f_{p_i} : i = 1, n\}$ est l'intensité au point p_i dans le volume π dont le nombre de points est n .

Pour l'hypothèse H_1 , la fonction de vraisemblance L_1 est déterminée par :

$$L_1 = (2\pi\sigma^2)^{-\frac{n_1+n_2}{2}} \prod_{p_i \in \pi_1; i=1}^{n_1} \exp\left\{ -\frac{1}{2\sigma^2} (f_{p_i} - c_1)^2 \right\} \prod_{p_j \in \pi_2; j=1}^{n_2} \exp\left\{ -\frac{1}{2\sigma^2} (f_{p_j} - c_2)^2 \right\}. \quad (3)$$

où $\{f_{p_i} : i = 1, n_1\}$ (resp. $\{f_{p_j} : j = 1, n_2\}$) est l'intensité au point p_i (resp. p_j) dans le sous-volume π_1 (resp. π_2) dont le nombre de points est n_1 (resp. n_2).

Le choix d'une de ces deux hypothèses est basé sur l'optimisation d'un critère qui est défini comme le rapport logarithmique de deux fonctions L_1 et L_0 :

$$\phi = \ln\left(\frac{L_1}{L_0}\right).$$

Le critère ϕ ne dépend que de deux types de paramètres : (a) les paramètres géométriques ρ qui déterminent la portion de plan modélisant un ECM et (b) les paramètres statistiques α qui déterminent les valeurs moyennes (c_0, c_1, c_2) des intensités. L'hypothèse H_1 est sélectionnée si la valeur maximale du rapport $\phi(\rho, \alpha)$ est supérieure à un seuil prédéterminé, c'est-à-dire :

$$\max_{\rho} \max_{c_1, c_2} \min_{c_0} \phi(\rho, \alpha) \geq S_{seuil}.$$

Sinon, l'hypothèse H_0 est retenue.

Les Masques De Convolution

En pratique, on prédéfinit un jeu de géométries G_i , $i = 1, \dots, m$ dans le domaine de variation possible des paramètres ρ . A une géométrie G_i donnée, la valeur maximale du rapport logarithmique de vraisemblance ϕ est calculée analytiquement par :

$$\phi_{max} = \frac{n_1 n_2}{2(n_1 + n_2)\sigma^2} (\hat{c}_1 - \hat{c}_2)^2. \quad (4)$$

où $\hat{c}_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} f_{p_i}$ et $\hat{c}_2 = \frac{1}{n_2} \sum_{j=1}^{n_2} f_{p_j}$. Plus formellement, si l'on considère la racine de ϕ , on obtiendra la formule finale d'estimation des éléments de contour en mouvement, qui est la convolution de la fonction intensité avec un masque tridimensionnel :

$$\left\{ \begin{array}{ll} \phi_{max} &= | \sum_{t=t_1}^{t_2} \sum_{p_t \in P_t} M(p_t) \bullet f(p_0 + p_t) |. \\ M(p_t) &= \frac{v_t}{\sigma} \left(\frac{n_2}{2n_1(n_1+n_2)} \right)^{0.5} & \text{si } p_t \in \pi_1. \\ M(p_t) &= -\frac{v_t}{\sigma} \left(\frac{n_1}{2n_2(n_1+n_2)} \right)^{0.5} & \text{si } p_t \in \pi_2. \end{array} \right. \quad (5)$$

où $f(\cdot)$ note les valeurs observées de l'intensité; P_t représente un jeu d'indices du masque $M(p_t)$ tel que $p_0 + p_t$ désigne tous les points dans le volume π centré au point p_0 ; n_1 (resp. n_2) est le nombre de points du volume π_1 (resp. π_2); Enfin, v_t désigne un coefficient compris entre 0 et 1, tenant compte de l'aspect numérique des images. Chaque pixel est considéré comme une petite région appelée *< maille >*. On calcule v_t selon la formule suivante :

$$v_t = |s_1 - s_2| / (s_1 + s_2). \quad (6)$$

où s_1 (resp. s_2) représente l'aire de la *< maille >* incluse dans π_1 (resp. π_2) à l'instant t .

Remarquons que dans le masque ainsi défini on a pris en compte en même temps deux éléments d'informations importantes :

1. L'information concernant la géométrie des contours, c'est-à-dire, la position et l'orientation.

2. L'information concernant la photométrie des contours, c'est-à-dire, la distribution des intensités lumineuses dans les deux régions adjacentes.

Nous utilisons ces masques dans notre algorithme pour déterminer les composantes de vitesse perpendiculaires aux contours. Ceci explique la raison pour laquelle notre approche est robuste.

Lissage Des Composantes Des Vitesses

Les deux motivations principales d'utilisation de cette opération sont :

1. Le mouvement observé dans le plan d'image est constitué en cas général de translation et de rotation. Si le déplacement d'une chaîne contour a une composante de rotation, il est impossible d'estimer directement les composantes de vitesse aux deux extrémités de cette chaîne contour.
2. Il est probable que certaines chaînes contours vont sortir en partie d'une image à un instant précis. Il est impossible d'estimer les composantes de vitesse sur l'ensemble de ces chaînes contours.

Notre idée pour effectuer le lissage consiste à d'abord calculer, pour une chaîne contour donnée, les valeurs moyennes M des composantes de vitesse sur cette chaîne et leurs variances V , puis à faire le lissage selon le critère suivant :

$$si \ ||v_x - M_x|| > V_x, \ v_x = M_x.$$

où v_x est une composante de vitesse à un pixel contour donné.

L'opération de lissage des composantes de vitesse permet à la fois de récompenser l'absence de détection dû au mouvement observé et aussi de corriger la fausse détection dû au bruit.

5 Mise En Correspondance Des Segments 2D

La modélisation de segment 2D à l'aide des coordonnées *plückeriennes* dans un espace 3D facilite :

1. l'estimation des paramètres dynamiques des segments 2D.
2. l'estimation des segments 3D correspondants.

Soit (\vec{m}, \vec{l}) les coordonnées *plückeriennes* d'un segment droit 2D, son équation s'écrit comme :

$$\vec{x} \times \vec{m} - \vec{l} = 0. \quad (7)$$

où \vec{x} est un point sur le segment. Il faut prêter attention à ce que les directions des trois vecteurs x, m et l respectent "la règle de main droite" dans un repère direct: OXYZ. La représentation d'un segment 2D par les coordonnées *plückeriennes* sera toujours unique sous la contrainte qu'un segment 2D se trouve sur le plan d'image.

Si le plan d'image de la caméra se situe au point $(0, 0, 1)$ dans le repère de la caméra, étant parallèle au plan oxy , on aura :

$$\begin{cases} \vec{m} = \{m_1, m_2, 0\} \\ \vec{l} = \{l_1, l_2, l_3\} \\ \vec{x} = \{x, y, 1\}. \end{cases} \quad (8)$$

En dérivant l'équation 7, on obtiendra :

$$\vec{\dot{x}} \times \vec{m} + \vec{x} \times \vec{\dot{m}} - \vec{\dot{l}} = 0. \quad (9)$$

où $\vec{\dot{x}}$ est la dérivée du point x et $(\vec{\dot{m}}, \vec{\dot{l}})$ sont les dérivées de (\vec{m}, \vec{l}) .

On peut décomposer le vecteur $\vec{\dot{x}}$ en deux composantes : celle parallèle au segment et celle perpendiculaire au segment, c'est-à-dire :

$$\vec{\dot{x}} = \vec{\dot{x}}_{\perp} + \vec{\dot{x}}_{\parallel}.$$

Puisque que $\vec{x}_{\parallel} \times \vec{m} = 0$, il est évident :

$$\vec{x} \times \vec{m} = \vec{x}_{\perp} \times \vec{m}.$$

l'équation 9 est devenue :

$$\vec{x}_{\perp} \times \vec{m} + \vec{x} \times \vec{m} - \vec{l} = 0.$$

De ce fait, on modélise un segment 2D dynamique par ses coordonnées *plückeriennes* avec leurs dérivées :

$$\{ \vec{m}, \vec{l}, \dot{\vec{m}}, \dot{\vec{l}} \}$$

Nous savons que la méthode d'extraction des contours spatio-temporels permet de déterminer à la fois la position des pixels contour et aussi le vecteur de déplacement perpendiculaire à chaque pixel contour correspondant. Connaissant un sous-ensemble des pixels contour spatio-temporels $\{ (\vec{x}, \vec{x}_{\perp})_i : i = 1, k \}$ qui forment un segment 2D dynamique sur le plan d'image, le meilleur estimé de paramètre $(\vec{m}, \vec{l}, \dot{\vec{m}}, \dot{\vec{l}})$ du segment 2D dynamique correspondant sera obtenu par optimisation de deux critères qui sont :

1. Critère 1 :

$$J_1 = \min \{ J(\vec{m}, \vec{l}) = \sum_{i=1}^k (\vec{x}_i \times \vec{m} - \vec{l})^2 \}. \quad (10)$$

avec $\|\vec{m}\| = 1$.

2. Critère 2 :

$$J_2 = \min \{ J(\vec{m}, \vec{l}) = \sum_{i=1}^k (\vec{x}_{i\perp} \times \vec{m} + \vec{x}_i \times \vec{m} - \vec{l})^2 \}. \quad (11)$$

avec $m_1 \bullet \dot{m}_1 + m_2 \bullet \dot{m}_2 = 0$.

La minimisation de ces deux critères revient à résoudre les équations suivantes :

$$\left\{ \begin{array}{l} l_1 = -m_2 \\ l_2 = m_1 \\ a \bullet m_2 - b \bullet m_1 - d \bullet l_3 = 0 \\ b \bullet m_2 - c \bullet m_1 - e \bullet l_3 = 0 \\ m_1 \bullet m_1 + m_2 \bullet m_2 = 1 \\ \dot{l}_1 = -\dot{m}_2 \\ \dot{l}_2 = \dot{m}_1 \\ a \bullet \dot{m}_2 - b \bullet \dot{m}_1 - d \bullet \dot{l}_3 + s \bullet m_2 - t \bullet m_1 = 0 \\ b \bullet \dot{m}_2 - c \bullet \dot{m}_1 - e \bullet \dot{l}_3 + p \bullet m_2 - q \bullet m_1 = 0 \\ m_1 \bullet \dot{m}_1 + m_2 \bullet \dot{m}_2 = 0 \end{array} \right. \quad (12)$$

où :

$$a = \sum_{i=1}^k x_i^2. \quad b = \sum_{i=1}^k x_i y_i. \quad c = \sum_{i=1}^k y_i^2.$$

$$d = \sum_{i=1}^k x_i. \quad e = \sum_{i=1}^k y_i. \quad s = \sum_{i=1}^k \dot{x}_i x_i.$$

$$t = \sum_{i=1}^k \dot{y}_i x_i. \quad p = \sum_{i=1}^k \dot{x}_i y_i. \quad q = \sum_{i=1}^k \dot{y}_i y_i.$$

Après quelques développements mathématiques, nous trouvons la solution :

$$\left\{ \begin{array}{l} m_1 = \frac{a \bullet e - b \bullet d}{L} \\ m_2 = -\frac{c \bullet d - b \bullet e}{L} \\ l_1 = -m_2 \\ l_2 = m_1 \\ l_3 = \frac{b^2 - a \bullet c}{L} \\ \dot{m}_1 = \frac{A \bullet e - B \bullet d}{M} \bullet m_2 \\ \dot{m}_2 = -\frac{A \bullet e - B \bullet d}{M} \bullet m_1 \\ \dot{l}_1 = -\dot{m}_2 \\ \dot{l}_2 = \dot{m}_1 \\ \dot{l}_3 = \frac{N}{M} \end{array} \right. \quad (13)$$

avec :

$$\begin{cases} L &= \sqrt{(a \bullet e - b \bullet d)^2 + (c \bullet d - b \bullet e)^2} \\ A &= s \bullet m_2 - t \bullet m_1 \\ B &= p \bullet m_2 - q \bullet m_1 \\ N &= (B \bullet a - A \bullet b) \bullet m_1 + (B \bullet b - A \bullet c) \bullet m_2 \\ M &= (a \bullet e - b \bullet d) \bullet m_1 - (c \bullet d - b \bullet e) \bullet m_2 \end{cases}$$

Si la distance focale f_c de la caméra n'est pas égale à 1, il faut seulement modifier le vecteur l et \dot{l} , c'est-à-dire :

$$\begin{cases} l_1 &= -m_2 \bullet f_c \\ l_2 &= m_1 \bullet f_c \\ \dot{l}_1 &= -\dot{m}_2 \bullet f_c \\ \dot{l}_2 &= \dot{m}_1 \bullet f_c \end{cases}$$

6 Suivi Des Indices Des Segments 2D

Nous avons résolu le problème concernant la mise en correspondance des segments 2D entre deux images consécutives. Nous voulons par la suite essayer d'établir les liens entre les indices des segments 2D dans une séquence d'images.

Sur chaque image, on connaît à la fois les segments 2D estimés à l'instant courant mais aussi les segments 2D prédits à l'instant précédent. Dans le cas idéal où il n'y a pas d'erreurs d'estimation, ces deux ensembles des segments 2D sont superposés parfaitement. De ce fait, la mise en correspondance entre les segments estimés et les segments prédits sur une même image consiste à trouver le plus proche voisin pour chaque segment parmi ces deux ensembles. La figure 4 montre la relation de correspondance que nous allons établir entre les segments 2D dans une séquence d'images.

Etant donné un segment 2D dynamique, les paramètres connus sont :

1. Ses coordonnées *plückeriennes* avec ses dérivés :

$$(m, l, \dot{m}, \dot{l}).$$

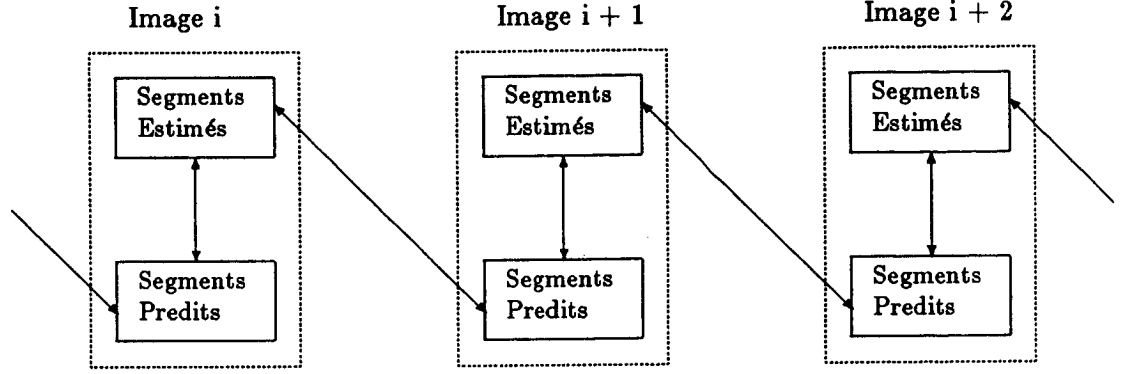


Figure 4 : La Correspondance Entre Les Segments 2D

2. Deux points d'extrémité du segment avec ses dérivés :

$$(p_1, p_2, \dot{p}_1, \dot{p}_2).$$

Si on paramètre un segment 2D dans un espace de dimension sept \mathbb{R}^7 qui est défini par les éléments :

$$C = \{m, \|l\|, \frac{1}{2}(p_1 + p_2)\}.$$

où le terme m impose une mesure sur l'orientation du segment, le terme $\|l\|$ impose une mesure sur la distance entre le segment et l'origine du repère et que le terme $\frac{1}{2}(p_1 + p_2)$ impose une mesure sur le centre de gravité du segment (voir la figure 5), l'ensemble des segments 2D peuvent être projeté vers un ensemble de points dans cet espace. La recherche du segment voisin pour un segment donné est devenu la recherche de deux points voisins dans l'espace \mathbb{R}^7 . Pour cette raison, nous employons une mesure de distance définie par l'équation suivante :

$$d(C_i, C_j) = \lambda_1 \text{acos}(\vec{C}_{im} \bullet \vec{C}_{jm}) + \lambda_2 \sqrt{\sum_{k=3}^5 (C_{ik} - C_{jk})^2} + \lambda_3 \sqrt{\sum_{k=6}^7 (C_{ik} - C_{jk})^2}.$$

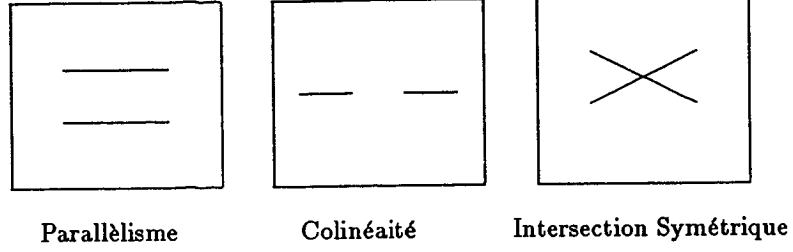


Figure 5 : Trois Mesures De Distance Entre Segments 2D

avec

$$\lambda_1 + \lambda_2 + \lambda_3 = 1.$$

où $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ sont les coefficients de pondération qui permet d'ajuster l'influence des trois mesures indiquées sur la mesure de distance.

Selon cette mesure, on va identifier deux segments voisins:

Soit $\{S_{i1} : i1 = 1, k\}$ l'ensemble des segments 2D estimés à l'instant t sur une image I et que $\{S_{j1} : j1 = 1, n\}$ l'ensemble des segments 2D prédits à l'instant $t - 1$ sur la même image I .

Deux segments 2D seront considérés comme étant identiques si :

$$\exists i1, j1 : d(S_{i1}, S_{j1}) \leq \text{seuil}. \quad (14)$$

où *seuil* est un seuil prédéterminé qui sert à identifier les segments 2D sortant de l'image précédente et entrant dans l'image I .

Etant donné un segment $i1$ parmi l'ensemble des segments 2D estimés, si on ne trouve pas son homologue dans $\{S_{j1} : j1 = 1, n\}$ (la mesure : $\min\{D(j1) = d(S_{i1}, S_{j1}), \forall j1\}$ est supérieure à *seuil*), le segment $i1$ est un segment entrant dans l'image I .

Par contre, étant donné un segment $j1$ parmi l'ensemble des segments prédits, si on ne trouve pas son homologue dans $\{S_{i1} : i1 = 1, k\}$ (la mesure : $\min\{D(i1) = d(S_{i1}, S_{j1}), \forall i1\}$ est supérieure à *seuil*), le segment $j1$ est un segment venant de sortir de l'image précédente.

7 Résultats Expérimentaux

Au cours du développement de notre projet de recherche, nous nous avons mis au point un système expérimental de vision dynamique intitulé VIDYR (Système de Vision Dynamique pour la Robotique). Les Modules principaux de VIDYR sont présentés à la figure 6 où on trouve :

1. Le noyau de VIDYR.

Sa fonction est de gérer la communication des données entre les modules et éventuellement l'utilisateur. Les données utilisées dans VIDYR sont généralement : (a) les images, (b) les trajectoires, (c) les paramètres et (d) les résultats des traitements. Pour faciliter cette tâche, nous avons défini un jeu de structures de données adapté aux différents niveaux des primitives de la vision. Le noyau de VIDYR gère aussi l'intégration d'une nouvelle application.

2. L'interface graphique.

Le système VIDYR est développé sur la station SUN sous UNIX. Nous utilisons les utilitaires de SUNVIEW et de SUNCORE pour créer un environnement graphique destiné à l'interfaçage graphique entre les modules de VIDYR et l'utilisateur, et aussi à l'affichage des résultats de traitements (par exemple: l'image des contours, l'image des segments 2D/3D, etc).

3. VISYR.

Nous avons à notre disposition un simulateur de vision en mouvement intitulé VISYR (Vision Synthétique pour la Robotique) développé à l'IRISA. Ce simulateur a été conçu au sein de l'algorithme de "Lancé de Rayons" qui a pour rôle de synthétiser l'image d'une scène spécifiée par le langage LGRC. L'intégration de VISYR dans VIDYR permet de tester les algorithmes de vision avec des données synthétiques. Ceci est très important pour analyser rapidement la validité de certains algorithmes.

4. Le Robot Mobile.

Nous avons réalisé un banc expérimental de vision en mouvement. Ce banc est constitué d'un robot manipulateur AID sur le poignet duquel

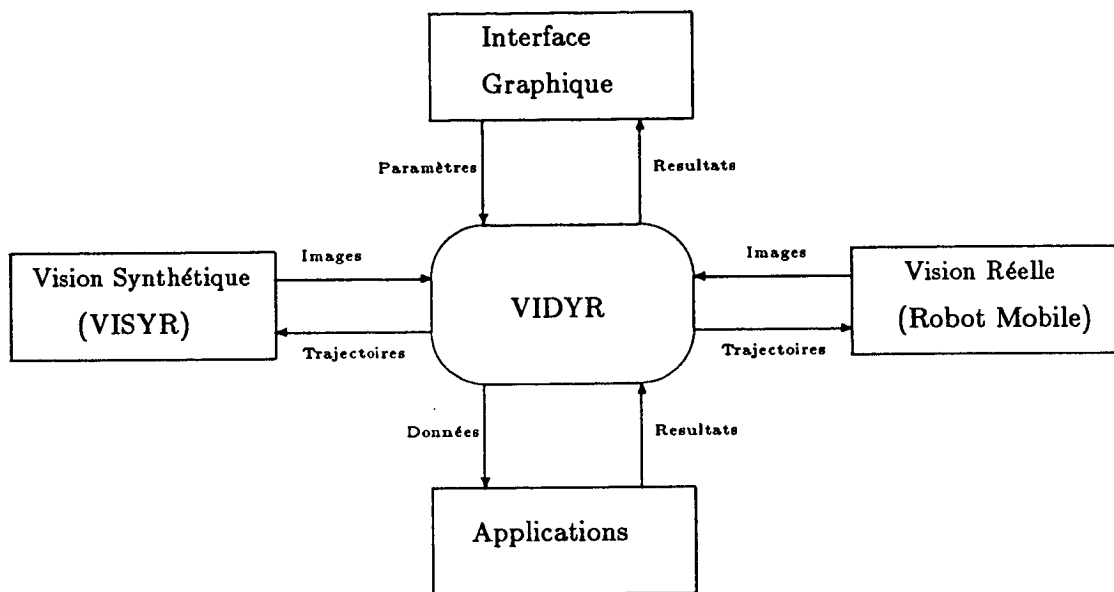


Figure 6 : Description De VIDYR

est installée une caméra CCD calibrée de marque MICAM. Le manipulateur est sous contrôle d'un processeur de traitement d'images EDIXIA. Ce dernier a une liaison directement avec la station SUN. Le système VIDYR peut commander facilement le robot en vue d'acquérir une séquence d'images de résolution 256×256 pour une tâche robotique donnée.

5. L'application.

Tous les algorithmes d'applications sont développés au sein de VIDYR. La programmation est basée sur le langage C. En fait, VIDYR représente un outils de recherche pour le développement des algorithmes de vision.

Plus précisément, nous avons utilisé VIDYR pour réaliser le travail concernant la reconstruction d'une scène 3D polyédrique. La figure 7 montre les modules de traitements principaux dans ce contexte.

En tant que module de traitement, l'algorithme de mise en correspondance des segments 2D décrits dans ce rapport a été réalisé au sein de VIDYR. Nous avons testé notre algorithme avec des images réelles. L'expérimentation nous indique que notre algorithme est très robuste. Nous montrons à la fin de ce rapport une série de résultats.

8 Conclusion et Discussion

Nous avons étudié dans ce rapport un algorithme efficace pour la mise en correspondance des segments 2D le long d'une séquence d'images. Notre approche possède deux avantages incontestables :

1. une meilleure qualité.
nous pouvons localiser fidèlement les pixels contours sur la première image quelle que soit la nature de l'image. La chaînage des contours nous permet d'avoir une meilleure estimation des orientations des contours et ainsi nous permet de choisir un jeu des masques le plus correct possible sans imposer aucune influence sur le temps de calcul. De plus, nous avons employé un masque optimal pour détecter les composantes de vitesse des contours, qui garantit un meilleur résultat de notre algorithme.
2. une plus grande rapidité.
le temps de calcul de notre algorithme est égale presque à une constante quel que soit le nombre des masques prédéfinis, qui correspond à-peu-près au temps de calcul nécessaire pour extraire les contours statiques sur une seule image (on ne compte pas le temps de calcul consommé par le chaînage).

Nous avons utilisé les coordonnées *plückeriennes* pour modéliser un segments 2D. Ceci permet une implantation directe de l'algorithme de l'estimation récursive des segments 3D.

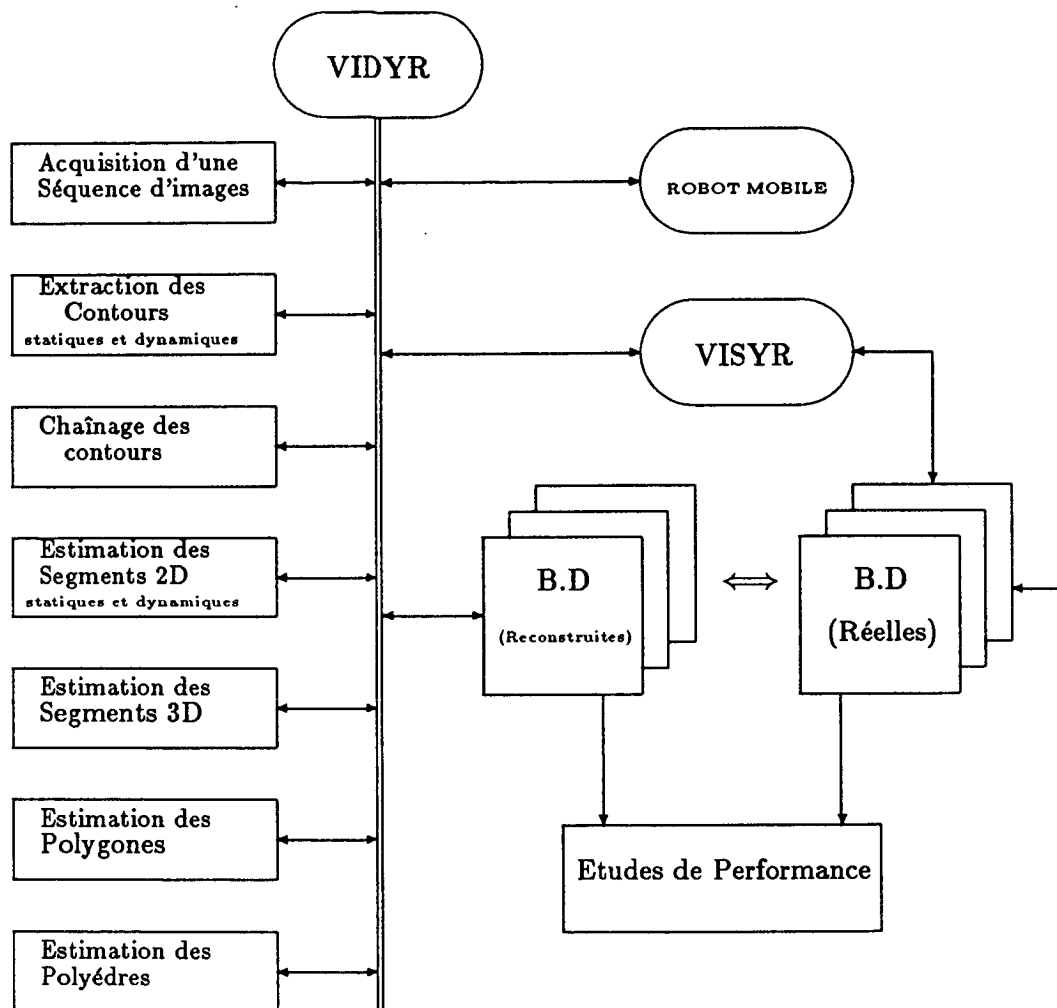


Figure 7 : Application de VIDYR pour la Reconstruction d'une Scène 3D Polyédrique

REMERCIEMENT

Ce travail s'inscrit dans le cadre du projet ORASIS: PRC communication Homme-Machine. Il a été effectué au sein de l'équipe TEMIS à l'IRISA. L'auteur tient à remercier P.BOUTHEMY pour les discussions enrichissantes qu'il a bien voulu lui accorder.

Bibliographie

- [1] BALLARD.D.H : " Generalizing The HOUGH Transform to Detect Arbitrary Shapes ", *Pattern Recognition Vol.13, No.2* , pp.111-122, 1981.
- [2] BOISSONNAT.J.D, O.D.FAUGERAS and E.LEBRAS : "Représentation des donnée stéréo par la triangulation de Delaunay", *Congrès AFCET-RFIA*, Antibes/FRANCE, Novembre, 1987.(in french)
- [3] BORGNE.M.L : " QUATERNIONS ET CONTROLE SUR L'ESPACE DES ROTATIONS ", *Rapport Intern de l'IRISA*, No.377, Octobre, 1987.
- [4] BOUTHEMY.P : " Estimation Of Edge Motion Based on Local Modelling ", *SPIE Vol.595, Computer Vision For Robotics* pp.162 - 169 , Cannes, December, 1985.
- [5] DERICHE.R : "Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector", *International Journal of Computer Vision*, P167-187,1987.
- [6] ESPIAU.B and RIVES.P : " Closed-Loop Recursive Estimation Of 3D Features For a Mobile Vision System ", *IEEE, International Conference on Robotics and Automation* March 30 - Aril 3 , Raleigh, 1987.
- [7] FREEMAN.H and DAVIS.L.S : "A corner-finding algorithm for chain-codes curves", *IEEE Trans. Comput.* 26(3), P297-303, 1977.

- [8] GIRAUDON.G : " Chaînage Efficace de Contour ", *Rapport de Recherche No.605*, centre SOPHIA ANTIPOLIS de l'INRIA, 1987.
- [9] HEGRON.G, ARNALDI.B and PRIOL.T : "VISYR : A simulation tools of synthetic vision for robotics", *Pro. of MARI87, Vo.II* , Paris, mai 1987.
- [10] KUROZUMI.Y and DAVIS.W.A : "Polygonal Approximation by the Minimax Method", *Computer Graphics and Images Processing 19* , 248-264, 1982.
- [11] LEU.J.G and CHEN.L : "Polygonal approximation of 2D shapes through boundary merging", *Pattern Recognition letters 7* , 231-238, 1988.
- [12] LOWE.D.G : "Three-Dimensional Object Recognition from Single Two-dimensional Images", *Artificial Intelligence 31* , 355 - 395, 1987.
- [13] PAVLIDIS.T : "SURVEY : A Review of Algorithms for Shap Analysis", *Computer Graphics and Images Processing 7*, 243-258, 1978.
- [14] RIVES.P : "Dynamic Vision: Theoretic Capability and Practical Problems" *Nato Workshop on Kinematic and Dynamic Issues in Sensor Based Control*, IL CIOCCO, TUSCANY Italy, October 25-31, 1987.

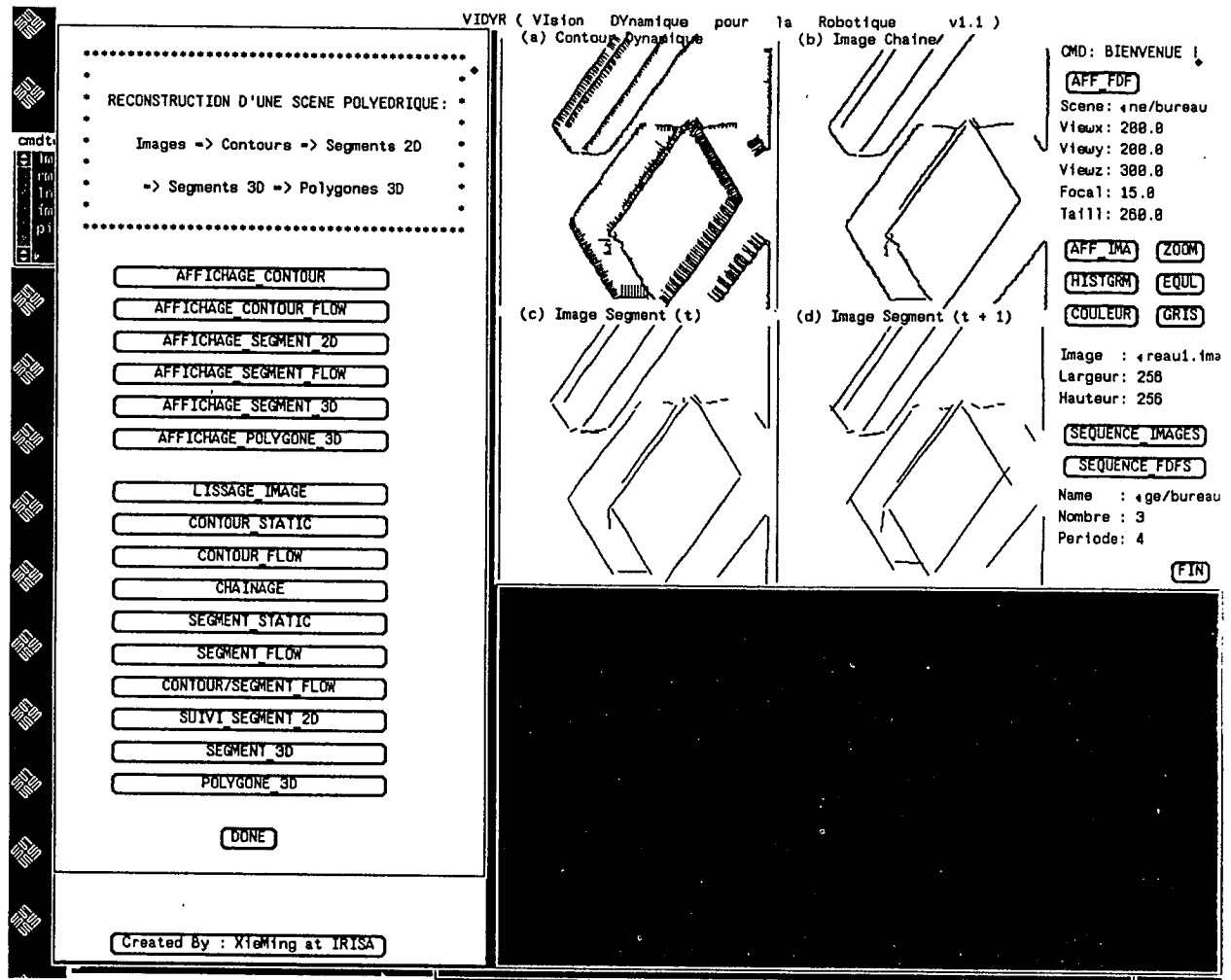


Figure 8 : L'image (a) montre les contours statiques avec leurs déplacements perpendiculaire aux contours. L'image (b) montre les chaînes obtenues par notre algorithme de chaînage des contours. L'image (c) montre les segments estimés 2D dans l'image courant. L'image (d) montre les segments 2D prédits dans l'image suivante

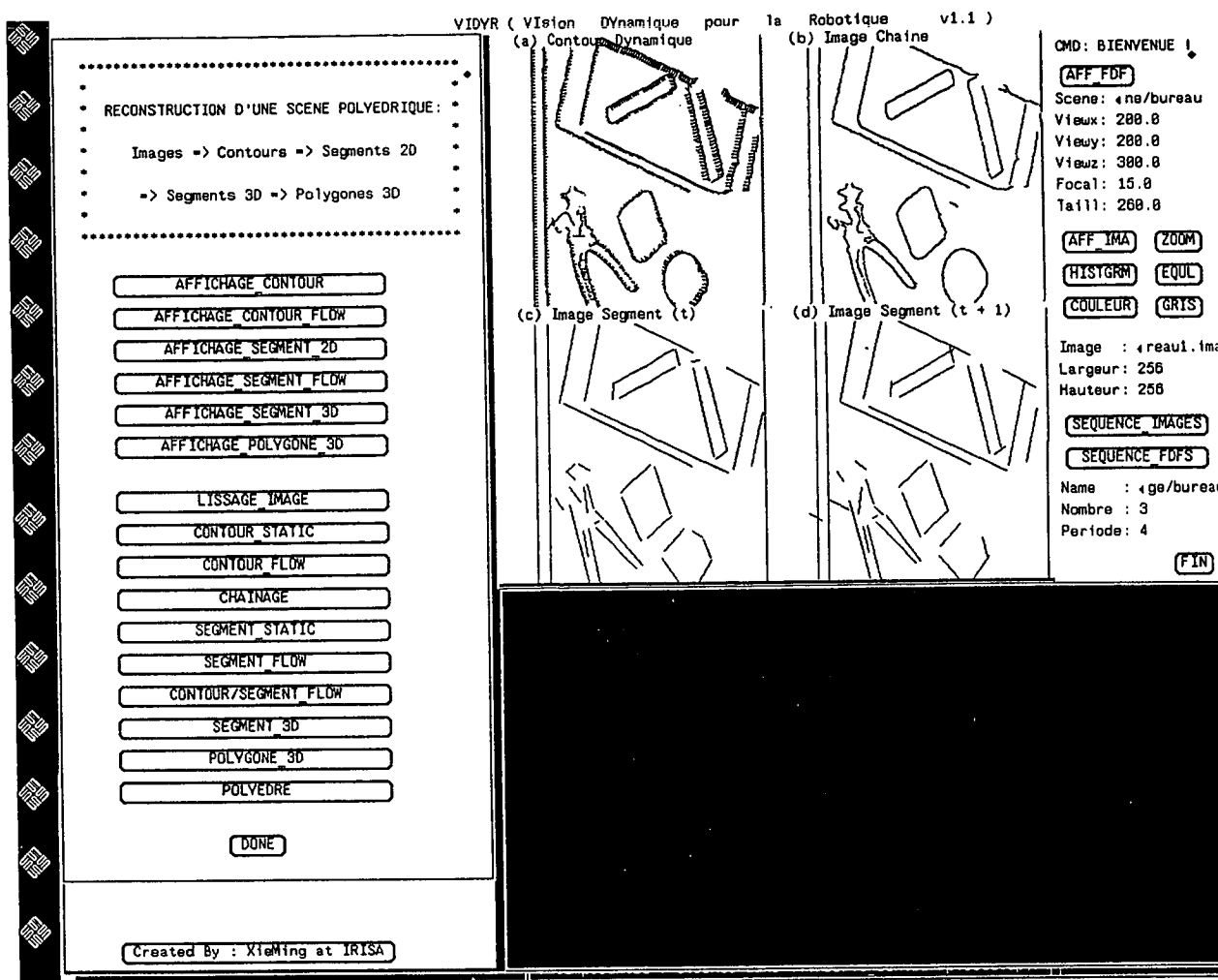


Figure 9 : L'image (a) montre les contours statiques avec leurs déplacements perpendiculaire aux contours. L'image (b) montre les chaînes obtenues par notre algorithme de chaînage des contours. L'image (c) montre les segments estimés 2D dans l'image courant. L'image (d) montre les segments 2D prédits dans l'image suivante

LISTE DES DERNIERES PUBLICATIONS INTERNES

- PI 425** **A MULTISCALE STOCHASTIC APPROACH FOR 2D INVERSE
PROBLEM IN CONDUCTIVITY**
Kenneth C. CHOU, Alan S. WILLSKY
40 Pages, Septembre 1988.
- PI 426** **CLASSIFICATION EN PRESENCE DE VARIABLES PREORDONNANCES
TAXONOMIQUES A CHOIX MULTIPLE. APPLICATION A LA STRUCTURA-
TION DES PHLEBOTOMES DE LA GUYANE FRANCAISE**
Israël-César LERMAN
184 Pages, Septembre 1988.
- PI 427** **IDENTIFICATION DES POLYGONES 3D A PARTIR DES SEGMENTS 3D**
Ming XIE, Patrick RIVES
46 Pages, Septembre 1988.
- PI 428** **UN MECANISME DE RESOLUTION POUR LE SYSTEME Q**
Christophe BOUCHY
62 Pages, Septembre 1988.
- PI 429** **ESTIMATION DES SEGMENTS 2D : UN ALGORITHME ROBUSTE**
Ming XIE, Patrick RIVES
34 Pages, Septembre 1988.

